(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification⁷: G06F 9/00

(21) International Application Number: PCT/US01/31157

(22) International Filing Date: 3 October 2001 (03.10.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/678,549   4 October 2000 (04.10.2000)   US

(71) Applicant *(for all designated States except US)*: INTEL CORPORATION [US/US]; 2200 Mission College Boulevard, Santa Clara, CA 95052 (US).

(72) Inventor; and
(75) Inventor/Applicant *(for US only)*: CORDOVA, Rois [US/US]; 5120 South Camellia Drive, Chandler, AZ 85247 (US).

(74) Agents: TROP, Timothy, N. et al.; Trop, Pruner & Hu, P.C., 8554 Katy Freeway, Suite 100, Houston, TX 77024 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

(54) Title: PEER TO PEER SOFTWARE DISTRIBUTION SYSTEM
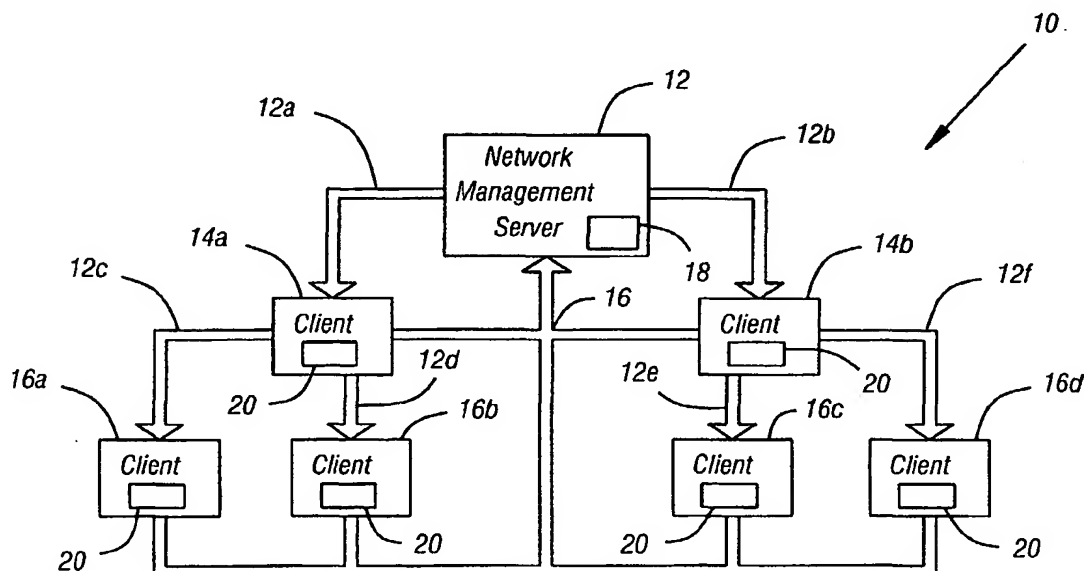
(57) Abstract: Software packages may be distributed to a plurality of clients in a network by initially distributing the software from a server to one or more clients. Those clients may then be enabled to forward the software package on to other clients contained in a list of addressee clients. Each client that receives the software package may then be called upon, in turn, to forward the software package to additional clients on the list. In this way, the load on the network management server, related to the distribution of software to a large number of clients in a network, may be reduced.

WO 02/29551 A2

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

PEER TO PEER SOFTWARE DISTRIBUTION SYSTEM

Background

This invention relates generally to the distribution of software to processor-based systems.

5　　　　Software may be distributed from a server to a variety of client processor-based systems. Conventionally, software packages may be periodically sent to a large number of processor-based systems within a network. For example, software updates, new software, application updates, and

10　operating system updates may be distributed from a server to a large number of clients. In distributed networks, it may be necessary to implement an automated system that distributes software to a large number of clients.

Distributing software updates to large numbers of

15　clients connected over a network may dramatically tax server resources. The time and resources devoted to distributing the software to a number of clients may dramatically increase the cost of the software update process. This is particular so when many of the clients may be accessible

20　through relatively low speed connections. For example, with set-top boxes that are basically processor-based systems that use television receivers as displays, relatively lower power processor-based systems may be provided with limited bandwidth Internet connections. The software download speed

25　may be relatively slow. As a result, the server may be taxed with slow downloads to a relatively large number of processor-based systems.

Thus, there is a need for better ways to distribute software to a large number of processor-based systems.

30　　　　　　　　　Brief Description of the Drawings

Figure 1 is a schematic depiction of one embodiment of the present invention;

1

Figure 2 is a flow chart for software utilized by the network management server shown in Figure 1 in accordance with one embodiment of the present invention; and

Figure 3 is a flow chart for software utilized by the
5   clients shown in Figure 1 in accordance with one embodiment of the present invention.

Detailed Description

Referring to Figure 1, a software distribution system 10 may involve the distribution of software from a network
10   management server 12 to a large number of clients 14 and 16. Each client 14 or 16 may be a processor-based system such as a desktop computer, a set-top box, a processor-based appliance or other processor-based systems. While the embodiment shown in Figure 1 includes six clients for
15   illustration purposes, a very large number of clients may receive software through the system 10.

In one embodiment of the present invention, a binary tree software distribution system may be utilized. Initially, the network management server (NMS) 12
20   distributes a software package together with a list of addressees to the client 14a, over the path 12a, and the client 14b over the path 12b. Each of the clients 14 in a first tier of clients then authenticates the package by providing a message back to the server 12 over a back or
25   alternate channel 17 for example. The authentication process assures that only authorized software has been distributed to the client 14 and prevents an outside entity from attempting to disrupt the system 10.

Each client 14 may then distribute the received
30   software to two clients 16 such as the client 16a and 16b in the case of client 14a or the clients 16c and 16d in the case of the client 14b. Each of the clients 16 in turn

2

authenticates the software through the channel 17.
Thereafter, each of the clients 16 may in turn distribute
the software to additional clients (not shown).

    In this way, software may be distributed to clients
5   that install the software and then distribute a copy of the
software, together with a list of addressees, to two
additional clients. Each time a software receiving client
14 or 16 may delete its identifier from the list of
addressees, in one embodiment, so that the software is not
10   repeatedly distributed to the same clients.

    The addressee list may also be parseable. The list
provided to the client 14a may be half of the original list
and the list provided to the client 14b may be the other
half of the original list. The client 14a may then parse
15   the list in half again and provide half of the list to the
client 16a and half the list to client 16b. Alternatively,
the list may be pre-parsed into a plurality of sections, as
received.

    Each upstream, software distributing server or client
20   identifies a receiving client to receive the software
package by taking the next name on the list of addressees,
in one embodiment. Thus, each client 14 or 16 may remove
its own identifier from the list after receiving the list
from an upstream source. Each client 14 or 16 then directs
25   the software package and the list of addressees to the next
client on the list. In this way, the software package,
together with the addressee list, may be progressively
distributed to every client in the network.   ,

    Because of the distributed or peer to peer nature of
30   the software distribution system, the demands on the network
management server 12 are reduced. The relatively longer
time frame involved in downloading the software package is
offloaded to the clients 14 and 16. The server 12 may

3

merely provide the needed authentication to the ultimate package recipients.

The authentication may be implemented by causing a client 14 or 16 to provide an authentication code back to

5   the server 12, in one embodiment. If the authentication code is correct, the server 12 acknowledges that a bona fide software package was received.

In some cases, the authentication code may be successively encrypted. The code may also be modified in· a

10 known or predetermined fashion by each successive client to avoid theft by unauthorized parties. For example, a rolling code scheme may be used. The package may also be accompanied by a checksum to ensure the package was received correctly.

15 While a system 10 is illustrated in which each source provides a software package in a binary tree to two ensuing clients, a variety of other distribution techniques may be utilized. The number of clients that receive software from an upstream client is essentially unlimited. To the extent

20 that the number is increased, the load on a particular upstream client software distributor is increased. Eventually, the software distribution load may be noticeable by the involved client's user. Therefore, it may be advantageous to limit the number of clients to which any one

25 client distributes the software package.

The software for enabling the distribution of software packages may, in one embodiment of the present invention, be preloaded on each client 14 or 16. In one case, the entire system 10 is managed by a service provider that controls the

30 hardware and software provided on each client 14 or 16. Thus, the clients 14 and 16 may be provided with the necessary hardware and software (in addition to the software

agents that are capable of automatically distributing the
software).

In some embodiments of the present invention, when a
software package is received, it may be immediately loaded
5   if client resources are available to implement the loading
routine. In some embodiments of the present invention, a
client 14 or 16 may wait until a period of inactivity in
order to distribute the software to additional clients. For
example, by monitoring the client's Advanced Configuration
10   and Power Interface (ACPI) power states, such as the
processor power states, the client 14 or 16 may determine an
advantageous, low activity time to distribute the software.
In this way the software distribution is less noticeable·and
disruptive to the sending client's owner or user. See ACPI
15   Specification, Revision 1.0, December 22, 1996. In another
embodiment of the present invention, the software
distribution may be programmed to occur at night. As still
another embodiment of the present invention, the transfer
time may be selectable by the client's user or owner.

20   In one embodiment of the present invention, the
software package may include the software for the receiving
client 14 or 16 as well as the software that enables the
receiving client, in turn, to distribute the software to
ensuing addressees on the list of addresses. In such case,
25   not only does the client 14 or 16 receive a software package
such as a software update, it also receives the software to
perpetuate the software distribution system. In one
embodiment of the present invention, the software
distribution software may be extinguished from the client
30   .after being executed.

Turning next to Figure 2, the network management server
12 software 34 for distributing the software packages may be
stored on a storage device 18. Initially, the software 34

5

receives a software package together with a list of client addresses as indicated at block 36.   These addresses may be in a form recognized by the network, such as Transmission Control Protocol (TCP)/Internet Protocol (IP) ports and

5   addresses as one example.   The software package and the addressee list is then transported to some number of clients such as the clients 14a and 14b as illustrated in block 38. At diamond 40, the server 12 awaits an authentication request from each of the clients 14.   When the

10   authentication request is received, if the appropriate code is provided, the software package is authenticated, for example by sending an acknowledgement, as indicated in block 42.   After the required number of client addressees have received their authentication, the flow may end.

15   In some embodiments, each client 14 or 16 may notify the server 12 that software was successfully installed on the client 14 or 16.   The server 12 may then monitor the amount of time that elapses without receiving a successful installation notification.   If the amount of time becomes

20   excessive, the server may check to determine if there is a problem with software installation on a particular client 14 or 16.   If there is a problem, the server 12 may intervene to facilitate the installation of software on the client with the problem and on other clients 14 or 16 that may

25   ultimately encounter problems.

     Referring next to Figure 3, the client software 44 may be stored on a storage 20 associated with a client 14 or 16. The software 44 is responsible for handling the software package received from an upstream source and distributing

30   the software to one or more downstream receiving clients. Initially, the software package and the addressee list is received by a given client as indicated in block 46.   The

6

client authenticates the software with the network
management server 12, as indicated in block 48.

The software package may then be installed on the
client as indicated in block 50. Thereafter, the software
5   package is distributed to the next name on the list of
addressees as indicated in block 52. In addition, in some
embodiments, the client 44 may remove its name from the list
of addressees. In the case where a sending client provides
the addressee list to two receiving clients, the sending
10   client may split the list into two and provide half the list
to each of the two receiving clients.

While the present invention has been described with
respect to a limited number of embodiments, those skilled in
the art will appreciate numerous modifications and
15   variations therefrom. It is intended that the appended
claims cover all such modifications and variations as fall
within the true spirit and scope of this present invention.

What is claimed is:

1.   A method comprising:
forwarding a software package together with a list of addressees to a first processor-based system; and
enabling said first processor-based system to automatically forward said software package together with at least part of said list of addressees to a second processor-based system, said second processor-based system being on said list of addressees.

2.   The method of claim 1 including enabling said first processor-based system to install said software package on said first processor-based system, make a copy of said software package, and transmit said software package to said second processor-based system.

3.   The method of claim 2 including causing said first processor-based system to automatically authenticate said software package.

4.   The method of claim 3 including causing said second processor-based system to automatically authenticate said software package by sending a message to said first processor-based system.

5.   The method of claim 4 including forwarding said software package together with a checksum to enable the second processor-based system to confirm with the first processor-based system that the software package was received correctly.

6.   The method of claim 1 including causing said first
processor-based system forwards said software package to
said second and a third processor-based system.

7.   The method of claim 1 including encrypting the
5   software package for transmission between said first and
second processor-based systems.

8.   The method of claim 7 including changing the
encryption in a known fashion with each successive transfer
from one to the next processor-based system.

10   9.   The method of claim 1 including transferring said
software package together with software that enables said
second processor-based system to transfer said software
package to a third processor-based system.

10.   The method of claim 1 including enabling said
15 first processor-based system to forward said software
package to said second processor-based system during a low
activity time on said first processor-based system.

11.   An article comprising a medium storing
instructions that enable a processor-based system to:
20   forward a software package together with a list of
addressees to a first processor-based system; and
           enable said first processor-based system to
automatically forward said software package together with at
least part of said list of addressees to a second processor-
25 based system, said second processor-based system being on
said list of addressees.

12.   The article of claim 11 further storing
instructions that enable the processor-based system to
install said software package, make a copy of said software
package, and transmit said package to a first processor-
5   based system.

13.   The article of claim 12 further storing
instructions that enable the processor-based system to cause
said first processor-based system to automatically
authenticate said software package.

10   14.   The article of claim 13 further storing
instructions that enable the processor-based system to cause
said second processor-based system to automatically
authenticate said software package by sending a message to
said first processor-based system.

15   15.   The article of claim 14 further storing
instructions that enable said processor-based system to
forward said software package together with a checksum to
enable the first processor-based system to confirm that the
software package was received correctly.

20   16.   The article of claim 11 further storing
instructions that enable the processor-based system to
enable said first processor-based system to forward said
software package to a second and third processor-based
system.

25   17.   The article of claim 11 further storing
instructions that enable the processor-based system to
encrypt the software package for transmission.

10

18.   The article of claim 17 further storing ·
instructions that enable the processor-based system to
enable said second processor-based system to change the
encryption in a known fashion.

5   19.   The article of claim 11 further storing
instructions that enable the processor-based system to
transfer said software package together with software that
enables said first processor-based system to transfer said
software package to said second processor-based system.

10   20.   The article of claim 11 further storing
instructions that enable the processor-based system to
enable said first processor-based system to forward said
software package to a second processor-based system during a
low activity time on the first processor-based system.

15   21.   A system comprising:
          a processor-based device;
          a storage coupled to said processor-based device
storing instructions that enable the processor-based device
to forward a software package together with a list of
20   addressees to a first processor-based system and enable the
first processor-based system to automatically forward said
software package together with at least part of said list of
addressees to a second processor-based system, the second
processor-based system being on the list of addressees.

25   22.   The system of claim 21 wherein said device is a
server.

          23.   The system of claim 22 wherein said server is a
network management server.

                                11

24.   The system of claim 21 wherein said device is a
client.

25.   The system of claim 21 wherein said storage stores
instructions to automatically transfer the software package,
5   the list of addressees, and software to enable further
distribution of the software package to additional
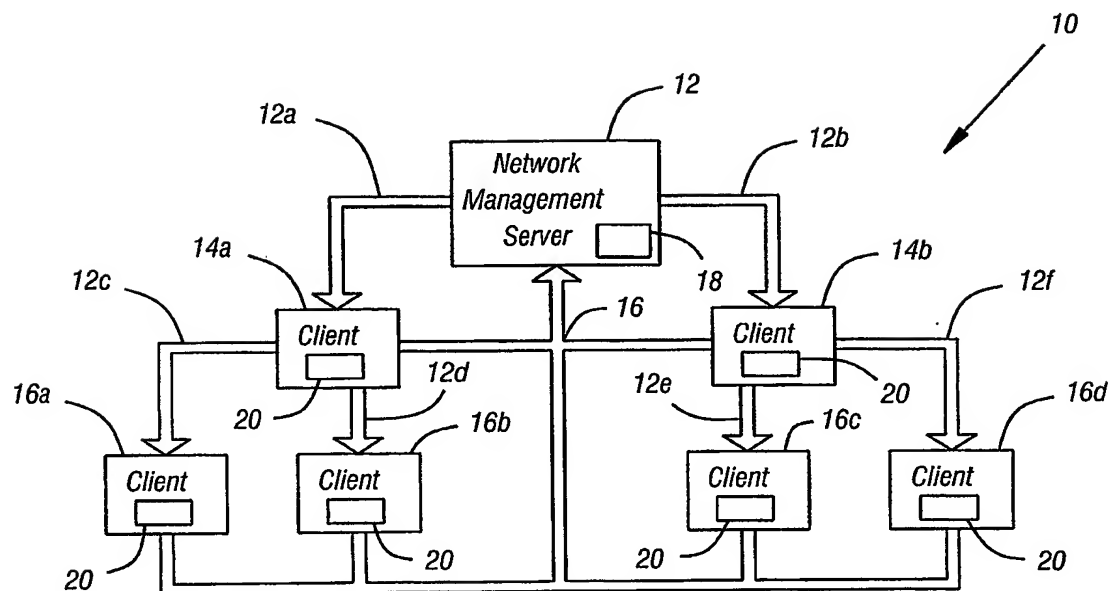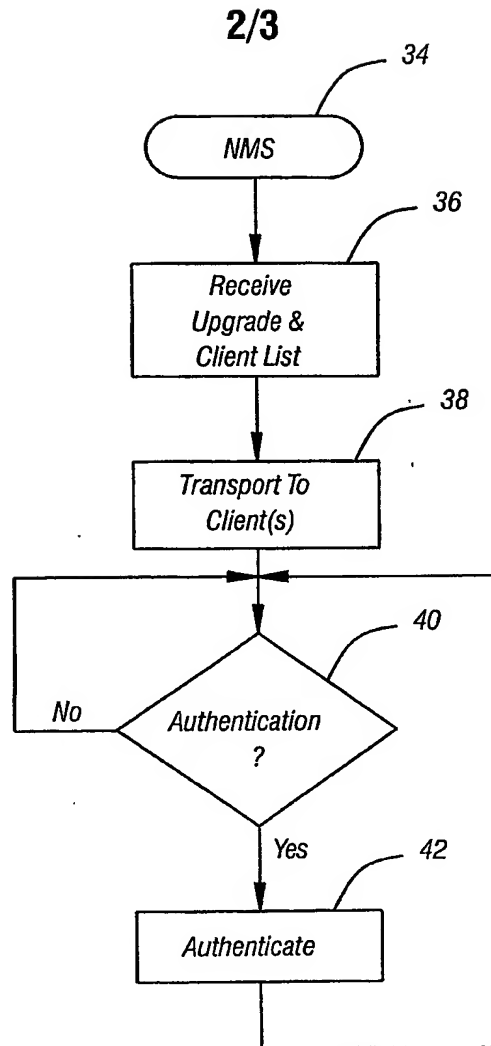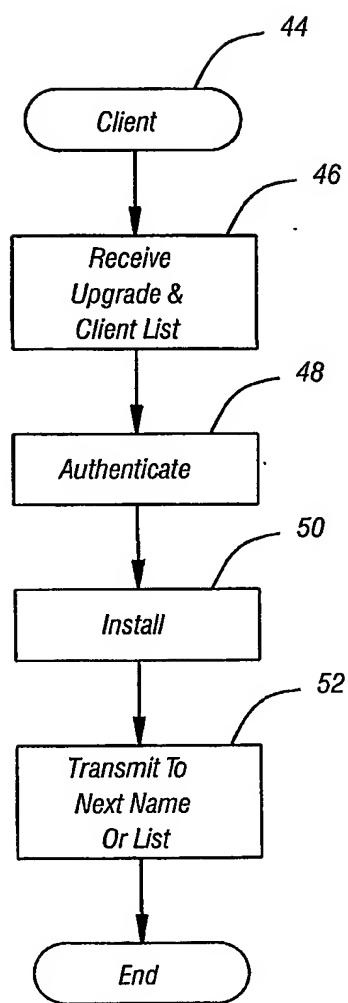processor-based systems.

1/3

FIG. 1

_34_

NMS

_36_

Receive
Upgrade &
Client List

_38_

Transport To
Client(s)

_40_

No ← Authentication
?

Yes _42_

Authenticate

FIG. 2

3/3

```
                        ┌──── 44
                    ╭─────────╮
                    │  Client │
                    ╰────┬────╯
                         │          ┌──── 46
                         ▼
                   ┌───────────┐
                   │  Receive  │
                   │ Upgrade & │
                   │Client List│
                   └─────┬─────┘
                         │          ┌──── 48
                         ▼
                   ┌───────────┐
                   │Authenticate│
                   └─────┬─────┘
                         │          ┌──── 50
                         ▼
                   ┌───────────┐
                   │  Install  │
                   └─────┬─────┘
                         │          ┌──── 52
                         ▼
                   ┌───────────┐
                   │Transmit To│
                   │ Next Name │
                   │  Or List  │
                   └─────┬─────┘
                         │
                         ▼
                    ╭─────────╮
                    │   End   │
                    ╰─────────╯
```

FIG. 3